REMARKS

The Examiner rejected claims 1-15, 17-18 and 20 under 35 U.S.C. §103(a) as allegedly being unpatentable over Andrew E. Ayers (U.S. Patent No. 5,857,105, and referred to as Ayers hereinafter) in view of Seema Hiranandani (U.S. Patent No. 5,812,855 and referred to as Hiranandani hereinafter).

The Examiner rejected claims 4 and 11 under 35 U.S.C. §103(a) as allegedly being unpatentable over Ayers in view of Seema Hiranandani, and further in view of Uma Mahadevan (U.S. Patent No. 5,797,013) and referred to as Mahadevan hereinafter).

Applicants respectfully traverse the §103 rejections with the following arguments.

35 U.S.C. §103(a): Claims): 1-15, 17-18 and 20

The Examiner rejected claims 1-15, 17-18 and 20 under 35 U.S.C. §103(a) as allegedly being unpatentable over Andrew E. Ayers (U.S. Patent No. 5,857,105, and referred to as Ayers hereinafter) in view of Seema Hiranandani (U.S. Patent No. 5,812,855 and referred to as Hiranandani hereinafter).

Claims 1-13 and 15

Applicant respectfully contends that claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, because Ayers in view of Hiranandani does not teach or suggest each and every feature of claims 1, 8, and 15.

As a first example of why claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "the callable procedure comprising a branch condition under which control flow code directs program flow from the branch condition to a code branch of two or more code branches, each said code branch being within the callable procedure and branching from the branch condition to program code within the callable procedure".

The Examiner alleges that Ayer's callee routine represents the callable procedure of claims 1, 8, and 15. In Ayers, FIG. 2, the routines T, U, and V are examples of such callee routines. Applicants contend that Ayers does not disclose that the callee routines satisfy the preceding feature of claims 1, 8, and 15.

For example, the callee routines do not comprise a branch condition as required by claims 1, 8, and 15. In fact, Ayers does not disclose what program code is comprised by the callee

routine other than the capability of being called. Ayers most certainly does not disclose that the callee routine comprises a branch condition.

In addition, the code branches in Ayers are branches from the caller routine to the callee routine are therefore not contained within the callee routine as required by claims 1, 8, and 15. See, e.g., code branches (1), (2), and (3) in Ayers, FIG. 2 from caller routine Z to callee routines T, U, and V, respectively.

Moreover, Ayers does not teach that the callee routine comprises "a branch condition under which control flow code directs program flow from the branch condition to a code branch of two or more code branches". For example in FIG. 2 of Ayers, the caller routine Z has control flow code for directing program flow to the code branches (1), (2), and (3). The callee routines T, U, and V are merely being called by the caller routine Z. Since the program flow is to the callee routine, and not from the callee routine, the callee routine does not and cannot direct program flow to the code branches (1), (2), and (3).

Applicant respectfully contend that the Examiner has incorrectly applied Ayers to the preceding feature of claims 1, 8, and 15. For example, the Examiner alleges that "directs program flow" is disclosed in Ayers by "converts an indirect call". However, in Ayers it is compiler code that converts the indirect call (see Ayers, col. 2, lines 35-36) and not the callee routine (which the Examiner alleges is the "callable procedure"). In contrast, claims 1, 8, and 15 require the callee routine to comprise the control flow code that "directs program flow", and Ayers does not disclose control flow code comprised by the callee routine.

As another example, the Examiner alleges that the branch condition is represented in Ayers as an a direct call, an indirect call, or in-line code. However, in Ayers it is the caller

routine that invokes the direct call, an indirect call, or in-line code. In contrast, claims 1, 8, and 15 require the callee routine to comprise the branch condition, and Ayers does not disclose a branch condition comprised by the callee routine.

As yet another example, a "branch condition" in a computer program, as understood by a person of ordinary in the art of computer programming, is a condition the analysis of which during execution of the computer program results in execution of a code branch of a plurality of code branches. Ayers does not disclose that the callee routine contains a branch condition.

Importantly, all of the details following "under which" in the preceding feature of claims 1, 8, and 15 pertain to the branch condition comprised by the callable procedure (alleged by Examiner to be Ayers' callee routine), and Ayers does not teach or suggest that the callee routine comprises any of said details. For example, Ayers most certainly does not disclose the specific constraint of "each said code branch being within the callable procedure and branching from the branch condition to program code within the callable procedure", and the Examiner has not demonstrated that Ayers discloses the preceding constraint.

In summary, the Examiner appears to have neglected pertinent distinctions between the compiler, the caller routine, and the callee routine, which relates to why the Examiner has incorrectly applied Ayers to the preceding feature of claims 1, 8, and 15.

As a second example of why claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new

procedures collectively comprise the two or more code branches "

The Examiner argues that Ayers teaches: "identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new procedures collectively comprise the two or more code branches; [Col. 2, Lines 51 - 59, see also Lines 35 - 50]."

In response, Applicant does not consider the citation of Ayers, col. 2, lines 35-59 to be an argument. The Examiner has not disclosed the Examiner's reasoning as to how Ayer, col. 2, lines 35-59 allegedly teaches the preceding feature of claims 1 8, and 15. Accordingly, Applicant maintains that the Examiner has not established a *prima facie* case of obviousness in relation to claims 1, 8, and 15.

As a third example of why claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure ...".

The Examiner argues that Ayers teaches: "recording a list of data entries [source listing, Col. 1, Line 7] corresponding to the respective new procedures [replaces, Col. 1, Line 10], each entry comprising a data item identifying the respective new procedure [executable object code, Col. 1, Line 9] ...".

In response, Applicant notes that the Examiner is asserting that a data entry in source code comprises a data item that identifies executable object code. Applicant maintains that the preceding assertion by the Examiner is incorrect, since source code, by definition, cannot

comprise or identify executable object code. As is well known in the art of computer software and computer programming, executable object code results from execution of a compiler, using source code as input.

As a fourth example of why claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "for the or each call statement, scanning the entries in said list to determine one for which there is correspondence between said branch conditions and **call parameters** directed to said control flow code by the call statement and modifying the call statement to replace the call to the original procedure by a call to the corresponding new procedure" (emphasis added).

The Examiner argues that Ayers teaches: for the or each call statement [caller, Col. 2, Line 36], scanning the entries in said list [source listing, Col. 1, Line 7] to determine one for which there is correspondence [a match, Col. 2, Line 44] between said branch conditions [indirect/ direct, Col. 2, Line 37 - 38] and **call parameters [characteristics, Col. 2, Line 44]** directed to said control flow code [caller's routine, Col. 2, Line 43] by the call statement [caller, Col. 2, Line 36] and modifying the call statement [reduces a number of indirect calls, Col. 1, Line 8 - 9] to replace [replaces, Col. 1, Line 10] the call to the original procedure by a call to the corresponding new procedure [inline listing/direct calls, Col. 1, Line 10]. [see also Col. 1, Lines 5 - 10, Col. 2, Lines 35 - 59, Col. 3, Lines 22 - 50, and Fig. 2]" (emphasis added).

In response, Applicant contends that the Examine has erroneously alleged that the "characteristics" discussed in Ayers, col. 2, line 44 are call parameters. Applicants maintain that Ayers, col. 2, line 44 most certainly does not teach or suggest that said "characteristics" are call

parameters.

As a fifth example of why claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "a code branch of two or more code branches".

The Examiner admits that "Ayers does not teach a code branches itself comprises two or more code branches".

The Examiner argues: "Hiranandani teaches a code branches itself comprises two or more code branches [see Fig. 5, see also Col. 1, Lines 1 - 67], for the purpose of performing optimization on procedures. [see Col. 11, Lines 59 - 60].... It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teaching of Ayers to include "a code branches itself comprises two or more code branches" for the purpose of performing optimization on procedures. [see Hiranandani, Col. 11, Lines 59 - 60]".

In response, Applicant notes that Hiranandani, col. 11, lines 59 - 67 recites: "The problem arises in attempting to perform IPA optimization on those procedures that are externally callable, such as procedure P1 508 and P2 514. Due to the uncertainty of the incoming parameter values, the procedures P1 508 and P2 514 cannot be optimized with respect to their parameters. Conventional IPA compilation systems cannot handle a call graph with unknown, or non-annotated, edges. Therefore, these conventional IPA systems are limited because they cannot process an incomplete call graph".

Applicant maintains that Hiranandani, col. 11, lines 59 - 67 does not support the Examiner's allegation that it would be obvious "to modify the teaching of Ayers to include "a

code branches itself comprises two or more code branches" for the purpose of performing optimization on procedures". In fact, Hiranandani, col. 11, lines 59 - 67 explains: "Conventional IPA compilation systems cannot handle a call graph with unknown, or non-annotated, edges. Therefore, these conventional IPA systems are limited because they cannot process an incomplete call graph".

Accordingly, the Examiner's argument for modifying Ayers by the alleged teaching of Hiranandani is not persuasive.

Based on the preceding arguments, Applicants respectfully maintain that claims 1, 8, and 15 are not unpatentable over Ayers in view of Hiranandani, and that claims 1, 8, and 15 are in condition for allowance. Since claims 2-6 depend from claim 1, Applicants contend that claims 2-6 are likewise in condition for allowance. Since claims 9-13 depend from claim 8, Applicants contend that claims 9-13 are likewise in condition for allowance.

In addition with respect to claims 5 and 12, Ayers and in view of Hiranandani does not teach or suggest the feature: "said method including optimising the or each new procedure for which a call parameter is a constant by propagating that constant through the new procedure."

The Examiner argues: "Regarding dependent claims 5 and 12, Ayers teaches, including optimizing the or each new procedure [reduces, Col. 5, Line 28] for which a call parameter is a constant [predetermined number, Col. 5, Line 27] by propagating that constant through the new procedure. [see also Col. 4, Line 51 - Col. 5, Line 59]".

In response, Applicant contends that the Examiner's argument is not persuasive, because

the "predetermined number" discussed in Ayers, col. 5, line 28 is not disclosed by Ayers as being a call parameter.

In addition with respect to claims 6 and 13, Ayers and in view of Hiranandani does not teach or suggest the feature: "said method including analysing a call statement, calling parameters and an associated new procedure to determine if they are compliant with predetermined **in-lining rules** and, if they are so compliant, replacing said call statement by a copy of the new procedure (emphasis added).

The Examiner argues: "Ayers teaches, ... including analyzing [comparing, Col. 2, Line 41] a call statement [caller, Col. 2, Line 43], [see also Col. 2, Lines 41 - 48] ... calling parameters [characteristic, Col. 2, Line 44] and an associated new procedure [callee rules [in-lining, Col. 2, Lines 53 - 55] and, [see Col. 2, Lines 35 - 58] ... if they are so compliant, replacing said call statement by a copy of the new procedure. [see Col. 1, Lines 5 - 10]".

In response, Applicant notes that Ayers, col. 2, lines 53-55 recites: "If the identified callee routine has already been **in-lined** (or there is a direct call present), it is executed and the program continues" (emphasis added). Applicants maintain that the Examiner has erroneously concluded, from the preceding quote of Ayers in which "in-lined" appears, that Ayers discloses the preceding feature of claims 6 and 13, which Ayers does not.

Claims 7, 14, 17-18 and 20.

As a first example of why claims 7, 14, 17-18 and 20 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature:

"wherein said one or more branching nodes and said respective code branches are contained within the callable procedure".

The Examiner argues that Ayers, col. 2, lines 35-59 teaches the preceding feature of claims 7, 14, 17-18 and 20.

In response, Applicants respectfully contend that Ayers, col. 2, lines 35-59 does not teaches the preceding feature of claims 7, 14, 17-18 and 20. As explained *supra* with respect in conjunction with claims 1, 8, and 15, Ayers does not disclose that the callee routine (which the Examiner alleges to be the callable procedure) contains one or more branching nodes as required by the preceding feature of claims 7, 14, 17-18 and 20.

As a second example of why claims 7, 14, 17-18 and 20 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "considering each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered".

The Examiner argues that Ayers teaches "considering each node in turn [sequentially accessed, Col. 4, Line 54] and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow

to the respective code branches is able to be represented as a function only of formal parameters [signature, Col. 5, Line 21] and global variables [Fig. 2], identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly [indirect/ direct, Col. 2, Lines 37 - 38] from the node being considered; [see Col. 4, Lines 51 - 65, Col. 5, Lines 10 - 31, and Col. 4, Lines 66 - Col. 5, Lines 9]".

In response, Applicant maintains that the Examiner's reference to "signature" in Ayers, col. 5, line 21 as being formal parameters is incorrect. Indeed, Ayers, col. 5, lines 19-21 recites: "Turning to FIG. 5, at the end of the profile match test, one or more prospective callee nodes have been identified which have passed both the **signature** match and profile match **tests**" (emphasis added). The Examiner has misinterpreted what Ayers discloses.

In further response, the Examiner's citation of Ayers [Col. 4, Lines 51 - 65, Col. 5, Lines 10 - 31, and Col. 4, Lines 66 - Col. 5, Lines 9] is not an argument that can be reasonably understood. The Examiner has not disclosed the Examiner's reasoning as to why the Examiner's citations in Ayers allegedly discloses the preceding feature of claims 7, 14, 17-18 and 20. Applicant has no idea of how the Examiner has reasoned that the [Col. 4, Lines 51 - 65, Col. 5, Lines 10 - 31, and Col. 4, Lines 66 - Col. 5, Lines 9] discloses the preceding feature of claims 7, 14, 17-18 and 20. Accordingly, Applicant maintains that the Examiner has not established a *prima facie* case of obviousness in relation to claims 7, 14, 17-18 and 20.

As a third example of why claims 7, 14, 17-18 and 20 are not unpatentable over Ayers in

view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "recording means for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch".

The Examiner argues that Ayers teaches "recording a list of data entries [source listing, Col. 1, Line 7] corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition; [Col. 1, Lines 5 - 10, and Col. 2, Lines 35 - 59]".

In response, Applicant contends that the Examiner's citation of Ayers [Col. 1, Lines 5 - 10, and Col. 2, Lines 35 - 59] is not an argument that can be reasonably understood. The Examiner has not disclosed the Examiner's reasoning as to why the Examiner's citations in Ayers allegedly discloses the preceding feature of claims 7, 14, 17-18 and 20. Applicant has no idea of how the Examiner has reasoned that the [Col. 1, Lines 5 - 10, and Col. 2, Lines 35 - 59] discloses the preceding feature of claims 7, 14, 17-18 and 20. Accordingly, Applicant maintains that the Examiner has not established a *prima facie* case of obviousness in relation to claims 7, 14, 17-18 and 20.

As a fourth example of why claims 7, 14, 17-18 and 20 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "for each said call statement, scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement;

and ... modifying the call statements to call said new procedures".

The Examiner argues that Ayers teaches "for each said call statement, scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement; and modifying the call statements to call said new procedures. [see Col. 1, Lines 5 - 10, Col. 2, Lines 35 - 59, see also Col. 3, Lines 22 - 50, and Fig. 2]".

In response, Applicant contends that the Examiner's citation of Ayers [Col. 1, Lines 5 - 10, Col. 2, Lines 35 - 59, see also Col. 3, Lines 22 - 50, and Fig. 2] is not an argument that can be reasonably understood. The Examiner has not disclosed the Examiner's reasoning as to why the Examiner's citations in Ayers allegedly discloses the preceding feature of claims 7, 14, 17-18 and 20. Applicant has no idea of how the Examiner has reasoned that the [Col. 1, Lines 5 - 10, Col. 2, Lines 35 - 59, see also Col. 3, Lines 22 - 50, and Fig. 2] discloses the preceding feature of claims 7, 14, 17-18 and 20. Accordingly, Applicant maintains that the Examiner has not established a *prima facie* case of obviousness in relation to claims 7, 14, 17-18 and 20.

As a fifth example of why claims 7, 14, 17-18 and 20 are not unpatentable over Ayers in view of Hiranandani, Ayers and in view of Hiranandani does not teach or suggest the feature: "two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node".

The Examiner admits: "Ayers does not teach for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program

flow is directed from the branching node."

The Examiner argues: "Hiranandani teaches for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node [see Fig. 5, Col. 1, Lines 1 - 67], for the purpose of performing optimization on procedures. [see Col. 11, Lines 59 - 60]."

In response, Applicant notes that Hiranandani, col. 11, lines 59 - 67 recites: "The problem arises in attempting to perform IPA optimization on those procedures that are externally callable, such as procedure P1 508 and P2 514. Due to the uncertainty of the incoming parameter values, the procedures P1 508 and P2 514 cannot be optimized with respect to their parameters. Conventional IPA compilation systems cannot handle a call graph with unknown, or non-annotated, edges. Therefore, these conventional IPA systems are limited because they cannot process an incomplete call graph".

Applicant maintains that Hiranandani, col. 11, lines 59 - 67 does not support the Examiner's allegation that it would be obvious "to modify the teaching of Ayers to include "a code branches itself comprises two or more code branches" for the purpose of performing optimization on procedures". In fact, Hiranandani, col. 11, lines 59 - 67 explains: "Conventional IPA compilation systems cannot handle a call graph with unknown, or non-annotated, edges. Therefore, these conventional IPA systems are limited because they cannot process an incomplete call graph".

Accordingly, the Examiner's argument for modifying Ayers by the alleged teaching of Hiranandani is not persuasive.

Based on the preceding arguments, Applicants respectfully maintain that claims 7, 14, 17-18 and 20 are not unpatentable over Ayers and in view of Hiranandani, and that claims 7, 14, 17-18 and 20 are in condition for allowance.

35 U.S.C.§103(a): Claims 4 and 11

The Examiner rejected claims 4 and 11 under 305 U.S.C. §103(a) as allegedly being unpatentable over Ayers, and in view of Seema Hiranandani, and further in view of Uma Mahadevan (U.S. Patent No. 5,797,013 and referred to as Mahadevan hereinafter).

Since claim 4 depends from claim 1, which Applicants have argued *supra* to not be unpatentable over Ayers under 35 U.S.C. §102(b), Applicants maintain that claim 4 is likewise not unpatentable over Ayers in view of Hiranandani and further in view of Mahadevan under 35 U.S.C. §103(a).

Since claim 11 depends from claim 8, which Applicants have argued *supra* to not be unpatentable over Ayers under 35 U.S.C. §102(b), Applicants maintain that claim 11 is likewise not unpatentable over Ayers in view of Hiranandani and further in view of Mahadevan under 35 U.S.C. §103(a).

In addition, Applicant respectfully contend that Ayers in view of Seema Hiranandani and further in view of Mahadevan does not teach or suggest the feature: "application of a cost-analysis algorithm based on predetermined rules about the length of the software".

The Examiner argues: "Ayers does not teach the application of a cost-analysis algorithm based on predetermined rules about the length of the software. Hiranandani teaches a code branches itself comprises two or more code branches [see Fig. 5, and Col. 1, Lines I - 67], for the purpose of performing optimization on procedures. Mahadevan teaches the application of a cost-analysis algorithm based on predetermined rules about the length of the software [Col. 10, Lines 46 - 52 & Lines 7 - 45] for the purpose of optimization. It would have been obvious to a

person of ordinary skill in the art at the time of applicant's invention to modify the teaching of Ayers to include "the application of a cost analysis algorithm based on predetermined rules about the length of the software" for the purpose of optimization. [see Hiranandani, Col. 10, Lines 46 - 52]".

In response, Applicant respectfully contends that Mahadevan, col. 10, lines 46-52 and 7-45 does not teach or suggest the application of a cost-analysis algorithm based on predetermined rules about the length of the software. The Examiner has misinterpreted Mahadevan.

In addition, the Examiner's argument is based on an improper modification of the secondary reference of Hiranandani. The Examiner argues that the primary reference of Ayers discloses various claimed features. The Examiner also argues that the secondary reference of Hiranandani has modified the primary reference of Ayers, by alleging that Hiranandani teaches or suggests optimization. The Examiner additionally argues that the secondary reference of Mahadevan has modified the secondary reference of Hiranandani, by alleging that "the application of a cost analysis algorithm based on predetermined rules about the length of the software" for the purpose of optimization. Applicants maintain that it is improper to argue that a claim feature is taught or suggested by a secondary reference through modification of another secondary reference. If the Examiner could modify a secondary reference in the preceding manner, then the Examiner would be able to show the existence of any element or feature of any claim merely by chaining a sufficient number of secondary references together in the preceding manner. Accordingly, Applicants respectfully maintain that the rejection of claims 4 and 11 under 35 U.S.C. §103(a) is improper and should be withdrawn.

Accordingly, Applicant maintains that the Examiner has not established a *prima facie* case

of obviousness in relation to claims 4 and 11.

CONCLUSION

Based on the preceding arguments, Applicants respectfully believe that all pending claims and the entire application meet the acceptance criteria for allowance and therefore request favorable action. If the Examiner believes that anything further would be helpful to place the application in better condition for allowance, Applicants invites the Examiner to contact Applicants' representative at the telephone number listed below. The Director is hereby authorized to charge and/or credit Deposit Account No. 09-0457.

Date: 12/19/2005

Jack P. Friedman
Registration No. 44,688

Schmeiser, Olsen & Watts
3 Lear Jet Lane, Suite 201
Latham, New York 12110
(518) 220-1850